Improving Gesture Recognition Efficiency with MediaPipe and YOLO-Pose

Nikita Andriyanov[1], Svetlana Mikhailova[1]
[1]Financial University under the Government of the Russian Federation

Keywords: Gesture Recognition, Keypoint Detection, Performance Algorithms, Computer Vision, MediaPipe, YOLO-Pose.

Abstract

This paper presents an improved combined approach for gesture recognition, combining a fast and lightweight keypoint detection algorithm using the MediaPipe method with a highly accurate YOLO-Pose model (integration of keypoints into the YOLO pipeline). This combination allows to drastically reduce the computational load compared to traditional convolutional networks while maintaining or even improving the recognition accuracy. As part of the extended study, in addition to the original experiment comparing different models on the HaGRID dataset, an additional experiment was implemented to evaluate the robustness of the system to changes in camera angle and gesture execution speed. The results show that the proposed method provides stable gesture recognition with mean Average Precision above 0.80 even under extreme conditions, which opens up prospects for its integration into mobile and embedded systems. We also tested different Artificial Intelligence ensembles to detect and classify gestures, but results for traditional methods are worse then YOLO-pose with MediaPipe.

1. Introduction

Gesture-based interfaces are increasingly in demand in a variety of fields, from augmented and virtual reality to disability assistance systems and smart home control. High interactivity without the need for physical contact makes gesture-based interfaces a flexible and intuitive means of interaction. However, to achieve an acceptable level of responsiveness and accuracy, many approaches require powerful hardware and significant computational resources, limiting their widespread adoption.

The goal of this research is to develop a gesture recognition method that combines the accuracy of state-of-the-art convolutional network-based detectors with the low computational cost of lightweight algorithms. Specifically, we integrate MediaPipe, a library optimised for mobile platforms and embedded systems, with Yolo-Pose, an extension of the classical YOLO model for predicting keypoint localisation. This solution allows to redistribute the workload: MediaPipe quickly and accurately extracts a set of keypoints, which are then used to build spatio-temporal features, while Yolo-Pose provides high-speed detection of bounding boxes and pose refinement. A schematic of the pipeline's operation is shown in Figure 1.

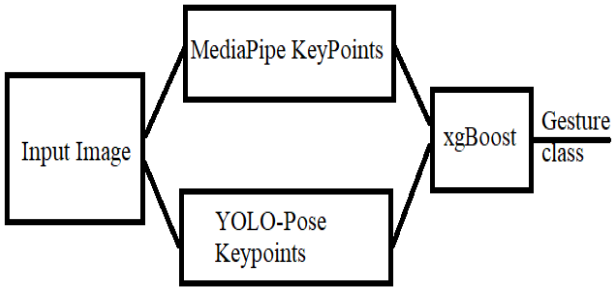

Figure 1. Gesture recognition pipeline architecture: keypoint detection (MediaPipe), spatio-temporal feature generation and classification (XGBoost).

2. Related works

Early approaches to gesture recognition were based on classical machine learning methods and simple features such as colour, texture and gradients (Mallat, 1989; Otsu, 1979). With the advent of deep convolutional neural networks (CNNs), automatic extraction of complex high-level features became possible (LeCun et al., 1998; Simonyan & Zisserman, 2014). Despite their high accuracy, such models require significant computational power and are not suitable for real-time systems on embedded devices.

The breakthrough came with the introduction of architectures for fast pose detection, such as OpenPose (Cao et al., 2019) and PersonLab (Papandreou et al., 2018), which translate the task to skeletally modelled data.These methods have significantly reduced the amount of raw data, but are still computationally intensive for mobile applications.

MediaPipe (Lugaresi et al., 2019) is a framework for building computer vision pipelines optimised for cross-platform and mobile devices. It is capable of real-time extraction of up to 33 key points of the hand with minimal latency.

The advent of YOLO-Pose (Dutta et al., 2022) allowed the integration of object detection and keypoint prediction into a single YOLOv8-based model. This provided accelerated performance while maintaining the accuracy typical of large CNNs. The combination of MediaPipe and Yolo-Pose leverages the strengths of each: MediaPipe quickly extracts the skeleton of the hand, while Yolo-Pose refines information about the context and position of the gesture in the frame (Lahiani et al., 2015; Zhang et al., 2023).

An additional area of optimization is CNN enhancement techniques, including pruning, quantization, and distillation (Andriyanov, 2022).

3. Materials and methods

We use the open HaGRID dataset (Kapitanov et al., 2022) containing 552,992 images of 18 gestures in FullHD resolution.

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLVIII-2-W9-2025-13-2025 | © Author(s) 2025. CC BY 4.0 License.

The images were captured under controlled studio conditions and in field scenarios, allowing for a variety of backgrounds, lighting and hand positions.

The HaGRID (Hand Gesture Recognition Image Dataset) (Kapitanov, 2022) was employed in this study to address the gesture recognition challenge. Developed by SBERDevices, the dataset comprises 552,992 RGB images categorized into 18 distinct gesture classes, with each class containing approximately 30,000 images to ensure balanced representation. Notably, 91% of the images are in FullHD resolution (1920×1080 pixels), capturing high-detail scenarios of real individuals performing gestures under diverse natural conditions. These include variations in lighting, camera distance (near to far), and hand positioning relative to the body, simulating real-world environments.

HaGRID emphasizes demographic diversity, featuring at least 34,700 unique participants aged 18–60 years, with a near-equal gender distribution (slightly skewed toward female representation). Each image is meticulously annotated with:
- bounding boxes identifying hands performing target gestures, labeled by gesture class;
- a leading_hand tag (right/left) to denote the dominant hand;
- “not gesture” annotations for hands in neutral or non-participatory positions.

To enhance robustness, the dataset accounts for multi-hand scenarios: hands not involved in the target gesture are explicitly annotated, mitigating false positives and improving model performance under partial occlusions or cluttered backgrounds.

An example of some of the gestures from the dataset is shown in Figure 2.

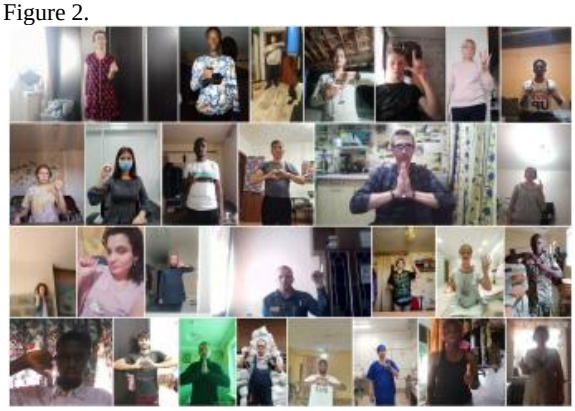

Figure 2. Demonstration of example gestures from the HaGRID dataset.

We proposed the original method based on pipeline from Figure 1 but extended using some new steps. So our method consists of the following steps:
Step 1. Pre-filtering of incorrect frames (missing keypoints or noise).
Step 2. Detection of key points of the hand using MediaPipe Hands Landmark pre-trained models.
Step 3. Detection of bounding boxes and pose refinement of the YOLO-Pose model.
Step 4. Feature generation for the XGBoost classifier for Mixture of keypoints:
1) Angles between skeletal hand segments (interphalangeal, wrist-wrist)

$$\theta_{ij,ik} = \arccos\left(\frac{(p_j - p_i)(p_k - p_i)}{\| p_j - p_i \| \times \| p_k - p_i \|}\right), \quad (1)$$

where p_i, p_j, p_k are coordinates of three consecutive joints (interphalangeal or wrist joints);

2) Ratio of areas of triangles formed by three key points

$$R_{(i,j,k)/(l,m,n)} = \frac{Area(p_i, p_j, p_k)}{Area(p_l, p_m, p_n)}, \quad (2)$$

where for ABC-triangle:

$$Area(A, B, C) = \frac{1}{2} \| (B - A) \times (C - A) \|; \quad (3)$$

3) Velocity characteristics of key point coordinate changes in a time window of N = 5 frames:

$$v_i(t) = \frac{p_i(t + \Delta t) - p_i(t)}{\Delta t}, \quad (4)$$

$$\mu_i = \frac{1}{N} \sum_{t=1}^{N} v_i(t), \quad (5)$$

$$\sigma_i^2 = \frac{1}{N} \sum_{t=1}^{N} (v_i(t) - \mu_i)^2. \quad (6)$$

Step 5. Training and validation of the XGBoost classifier on 80% of the data, testing on 20%.

Figure 3 shows recognizable gestures from the base.

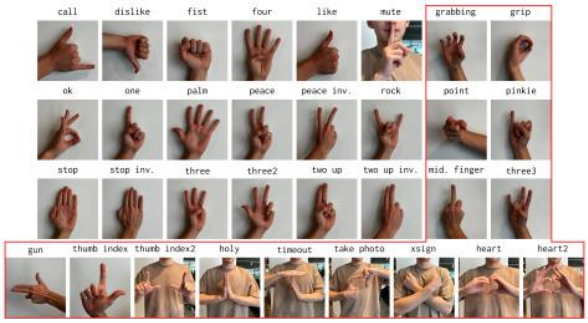

Figure 3. Examples of gestures

Feature extraction based on key points for gesture classification involves the application of spatial and temporal features to enhance the informativeness of the data. Spatial geometric features such as area of triangles, aspect ratio and angles help to analyze the physical features of gestures. For example, the change in area between key points can be used to recognize the “OK” gesture. Temporal dynamic features, including velocity and motion trajectories, can analyze kinematics and divide gestures into phases, which is critical for dynamic actions such as “swing”.

This approach provides invariance to illumination and texture, and increases interpretability since each feature has an obvious physical meaning. Hybrid features can be integrated into various machine learning models such as SVM or LSTM. Experimental results show that the use of geometric features improves classification accuracy:
Experimental results show that the use of geometric features improves classification accuracy.

Figure 4 shows a map of key points based on the MediaPipe library.

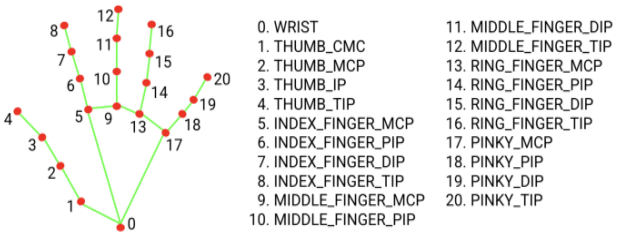

Figure 4. MediaPipe Arm Key Points

A similar pattern was used for the YOLO-pose markup.

ControlNet is an extension of the diffusion image generation model (e.g., Stable Diffusion) designed to control the visual output by supplying additional conditions in the form of maps or images such as pose maps, depth maps, contour images, normal maps, and other forms of structural information. The primary goal of ControlNet is to enable precise and controlled editing of the generated images, while maintaining the creative flexibility and generation quality inherent in diffusion models. The basic ControlNet architecture builds on the already trained text-to-image model of the Stable Diffusion type. The difference is that instead of a purely textual prompt, image generation also depends on a structural input called a conditioning map. This map can be a pose skeleton, a depth map, selected contours, etc. In the case of gesture recognition, it is most logical to use a pose map derived from the key points of the hand extracted by MediaPipe.

ControlNet uses a copy of the weights of the main Stable Diffusion U-Net block, trained separately to handle additional conditions.In this way, the U-Net model is split into two parallel threads: the main thread processes text-based latent representations and the auxiliary thread processes image-based representations with condition. Integration takes place by means of residual (residual) connections between layers.This preserves the original generative power of the model while adding controllability and accuracy.

To ensure the versatility of the architecture, a module called ‘Zero Convolution Layer’ is used - it is initialised with zeros, ensuring that prior to training, the impact of the new modality on the network is minimal.Subsequently, this module is trained by adapting to a given condition structure (e.g., pose) without violating the general properties of the underlying model. In this way, ControlNet effectively ‘adapts’ to an already trained model and does not require a complete re-training of all parameters, which speeds up development and makes it compatible with existing versions of Stable Diffusion.

A special feature of ControlNet is its high flexibility: the same text query can be interpreted differently depending on the shape of the input card.For example, the prompt ‘cybernetic hand’ can be visualised as an outstretched hand with ‘V’ shaped fingers if the corresponding pose is given as input, or as a palm touching the screen if the condition is an open-pose from MediaPipe.
For gesture-based image generation tasks, one of the most common ControlNet modes, pose, is used in the context of this paper.The key points extracted by MediaPipe (21 points on each hand) are connected into a skeletal structure, which is then visualised as a binary image (e.g. white lines on a black background). This map is fed into ControlNet at the same time as a textual cue that sets the overall context of the scene, for example:‘robotic arm reaching out of screen’, “superhero in action pose”, or “anime character casting spell with fingers”.

The integration of ControlNet into the gesture recognition pipeline not only enables precise control over the output of generative models but also unlocks new opportunities for multimodal interaction. Users can manipulate both the pose of a character and the stylistic or semantic content of the generated image using hand gestures. For example, a “thumbs up” gesture could trigger portrait generation, a pointing gesture could shift focus to a landscape scene, while a circular hand motion might initiate animation or a transitional visual effect within an interface.

Moreover, the ControlNet architecture exhibits high robustness to distortions and noise. Even in the presence of partial occlusion of keypoints (e.g., due to self-occlusion) or suboptimal hand poses, the system is capable of reliably interpreting the structural information, owing to its training on a large and diverse set of annotated data. This characteristic is particularly valuable in real-time applications where hand gestures may be captured from non-standard angles by a moving or fixed camera.

In practice, ControlNet can be deployed for gesture-based image generation either locally on a GPU, in the cloud, or via API services. Open-source implementations based on PyTorch and the Hugging Face diffusers library are available for local execution. The generation speed depends on the hardware configuration; however, even on consumer-grade GPUs (e.g., RTX 3060/3080), a 512×512 pixel image can typically be synthesized within 2–5 seconds, making the method suitable for interactive use.

In the context of this study, ControlNet serves as the final stage of the generation pipeline: once MediaPipe extracts the hand keypoints, they are converted into a pose map and passed, alongside a textual prompt, to ControlNet. The output is a synthesized image that visually corresponds to both the hand’s pose and the semantic context provided by the user. This transforms the system from a mere gesture classifier into a powerful multimodal interface, integrating computer vision, natural language processing, and generative modeling.
Overall, the architecture of ControlNet demonstrates strong potential for a wide range of applications, including creative design, visual programming, digital storytelling, and educational or game-based interfaces. Thanks to its modular design and compatibility with existing image generation pipelines, it can be readily adapted to various use cases, including gesture-based control, pose recognition, augmented reality, and beyond.

4. Results and Discussion

The mean Average Precision (mAP) metric (for gesture detection and classification in the video) was used as the main metric. The average frame rate per second (FPS) on the HaGRID test data was also measured.

To evaluate the performance of the proposed method, we compared the following variants:

- YOLOv12x - basic detection model without keypoint detection;
- MediaPipe + SVM;
- MediaPipe + decision tree;

- YOLOv12-pose + SVM;
- YOLOv12-pose + decision tree;
- YOLOv12x + MediaPipe + XGBoost combination.

Experiments were conducted on a workstation with an Intel Core i9-10900K processor (10 cores, 3.7-5.3 GHz), an NVIDIA RTX 3080 graphics accelerator (10 GB GDDR6X), 32 GB of DDR4 RAM and an NVMe SSD.

Table 1 shows the comparison results of different approaches in quality and performance.

Model	mAP	FPS (average)
YOLOv12x	0.82	12
MediaPipe+SVM	0.74	25
MediaPipe+Decision Tree	0.76	30
YOLOv12-pose+SVM	0.67	10
YOLOv12-pose+Decision Tree	0.69	11
YOLOv12x+MediaPipe+xgBoost	0.86	20

Table 1. Comparison of Gesture Recognition.

The poor results of YOLO-pose may be due to the undertraining of the model.However, we observe that the approach combining YOLO and MediaPipe improves the model quality by 4 percentage points. It should be noted that MediaPipe includes optimized models that can be implemented on CPUs.

The results demonstrate that the combined method achieves the highest mAP at an acceptable FPS level sufficient for real-time applications.

Furthermore extended experiment was produced. It investigates robustness to viewing angle and speed of gesture execution.

Purpose of the experiment include estimation of proposed method in different angles of view.
To test how varying the viewing angle (30°, 45°, 60°) and gesture execution speed (0.5, 1 and 2 gestures/s) affects recognition performance.

For each of the 18 gestures, subsets of 1000 images were generated for each pair of conditions (angle × speed). Total: 18 × 3 × 3 = 162 sets (162,000 images).

We also use mAP for each combination and estimate standard deviation of mAP within each gesture.

We use one-factor analysis of variance (ANOVA) to assess the statistical significance of the effect of parameters.

Table 2 shows mAP at different viewing angles and speeds.

Angle	Speed	mAP	StdDev
30°	Slow	0.88	0.02
30°	Normal	0.86	0.03
30°	Fast	0.83	0.05
45°	Slow	0.87	0.02
45°	Normal	0.85	0.03
45°	Fast	0.82	0.06
60°	Slow	0.86	0.03
60°	Normal	0.84	0.04
60°	Fast	0.80	0.07

Table 2. Robustness Investigation.

One-factor ANOVA confirmed a statistically significant effect of angle ($p < 0.01$) and velocity ($p < 0.01$) on mAP.

The analysis of Table 1 shows that among all tested methods the combined approach YOLOv12x + MediaPipe + XGBoost demonstrates the best accuracy (mAP = 0.86). This supports the hypothesis of the synergistic effect of combining easy and fast keypoint localisation with high performance classification. In addition, this method maintains an acceptable processing speed of about 20 frames per second, which allows it to be used in real-time tasks.

If we compare MediaPipe in combination with simple classifiers (SVM and decision tree), we see mAP at 0.74-0.76. This means that even without the YOLO component MediaPipe can extract quite informative features, especially in stationary conditions. However, they lose out to the more powerful XGBoost-based ensemble, indicating the importance of sophisticated analyses of spatio-temporal features, especially in the case of variable gestures.

The behaviour of the YOLOv12-pose model is also of interest. Despite its potential power in localising key points, the accuracy in combination with SVM and decision tree does not exceed 0.69. This may be due to the fact that the model is not adapted to the features of HaGRID annotations or that its built-in architecture is not so efficient without additional feature aggregation.

The analysis of Table 2, reflecting the model's robustness to external parameters, confirms stable performance even when the viewing angle and gesture speed are changed. The greatest decrease in accuracy occurs at an angle of 60° and fast speed (2 gestures/s), where mAP drops to 0.80. This is logical: at high angles, some key points may become invisible (self-occlusion), and high speed leads to image blurring or incomplete gesture capture. However, a decrease of 6 percentage points is considered moderate and acceptable for most practical applications.

It can also be seen how accuracy decreases with increasing speed and angle. The smoothness of the mAP decline confirms that the model is not retrained for specific conditions and is able to generalise to variable situations.

Additionally, statistical analysis (ANOVA) showed that both factors - angle of view and speed of movement - have a statistically significant effect on accuracy, but their effect is additive, i.e. the model degrades smoothly rather than abruptly, which also speaks in favour of its stability.

Thus, the proposed method has a balance between accuracy, stability and performance, which makes it applicable in a variety of scenarios from industrial interfaces to user applications. and perspectives

Figure 5 shows the results of processing different gestures using proposed method.

The MediaPipe + YOLO-Pose + XGBoost model can interpret a specific gesture as one of the conditional commands to generate an image. For example, Table 3 shows different variants for using it in ControlNet (L. Zhang, 2023).

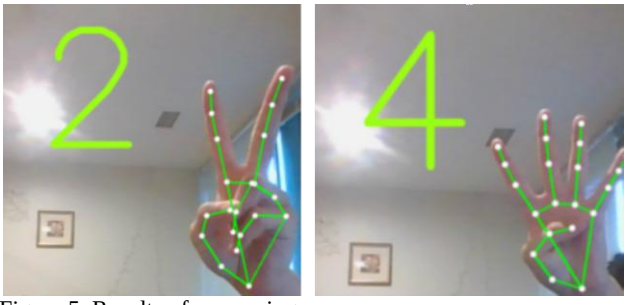

Figure 5. Results of processing

Hand Gesture	Interpretation	Input for ControlNet
Fingers in a ‘V’ shape	‘create landscape’	prompt = "mountain view"
Palm Up	‘add light’	conditioning = brightness mask
Fist	‘generate portrait’	pose + face prompt

Table 3. Interaction with ControlNet.

Further, based on the recognised gestures, interaction with image generation systems, e.g. using ControlNet, can be implemented. The key points of the hand extracted using MediaPipe can be converted into a pose map, a structure that is used as a control input to the ControlNet model. Such a map can be constructed by visualising 21 key points and connecting them with lines to create a skeletal representation of the hand on a black and white background. ControlNet supports various conditional control modes, including pose and openpose, which makes integration particularly convenient.

In the next step, the visual position of the hand (e.g. an outstretched palm or a clenched fist) can be interpreted as not only a command, but also as a pose to generate an image. This allows unique scenes to be generated in combination with a textual description. For example, the combination: Gesture → skeleton → ControlNet image together with the prompt ‘cybernetic hand reaching out’ creates an image of a character with an outstretched arm. This control gives you more freedom and visual accuracy when generating images.

An example of a practical scenario could be as follows. A camera captures a user demonstrating a certain gesture. This gesture is categorised as a ‘create character with wings’ command. The pose is then formed from key points, visualised and fed to ControlNet as a conditioning image. In combination with the text prompt, the Stable Diffusion model creates an image of a character with the desired body configuration - for example, an outstretched arm or flight. This sets the stage for the use of gesture recognition in creative interfaces and VR/AR scenarios.

The advantages of this approach are clear: it requires no keyboard input, scales easily to different devices, and can provide intuitive real-time interaction, making the image generation process more visual, interactive, and adaptive to user gestures. between accuracy, robustness, and performance, making it applicable in a variety of scenarios from industrial interfaces to custom applications. and perspectives.

Figure 6 demonstrates ControlNet application using gestures.

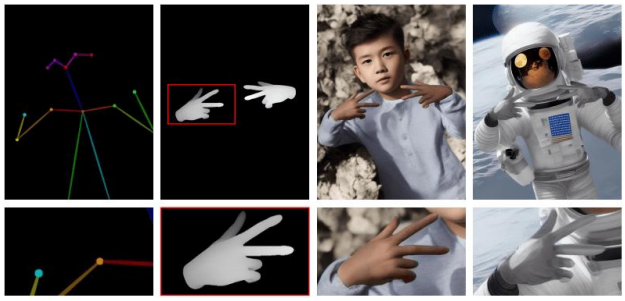
Figure 6. Example of image generation using gestures

5. Conclusions

In this work, a combined method for gesture recognition was proposed and experimentally validated, integrating MediaPipe, Yolo-Pose, and the XGBoost classifier. This approach achieved high accuracy (mAP = 0.86) with acceptable processing speed (around 20 frames per second), making it suitable for real-time systems. Unlike traditional CNN models, the proposed architecture is effective even on consumer-grade hardware and maintains quality across a variety of background conditions and hand positions.

Additionally, experiments demonstrated the method's robustness against variations in viewing angles and hand movement speeds. Even under unfavorable conditions (60° angle, high speed), the accuracy remained at mAP = 0.80. This result confirms the model's ability to generalize and its applicability in practical scenarios—ranging from user interfaces to robotic systems and VR/AR environments. By utilizing key points and analytical features, the system shows resilience to shifts, partial occlusions, and noise.

Particular attention was given to the integration of recognized gestures with the image generation system ControlNet. It was shown that hand key points can be used as input structures for the generative model, allowing users to specify not only commands but also visual representations. This expands the scope of application from recognition to multimodal interaction, encompassing computer vision and generative neural networks. In the future, this approach could serve as a foundation for creating intuitive and adaptive interfaces in creative, educational, and interactive applications.

Acknowledgements

The research was funded by RSF, Project №25-21-20028.

References

Andriyanov, N.A., Dementiev, V.E., Tashlinskiy, A.G., 2022. Development of a Productive Transport Detection System Using Convolutional Neural Networks. *Pattern Recognit. Image Anal.* 32, 495–500.

Cao, Z., Hidalgo, G., Simon T., Wei S., Sheikh Y., 2019. OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. *IEEE TPAMI*, 43(1), 172–186.

Dutta, A., Maji, D., Nagori, S., Mathew, M., Poddar, D., 2022. YOLO-Pose: Enhancing YOLO for Multi Person Pose Estimation Using Object Keypoint Similarity Loss. *arXiv:*2204.06806, 1-10.

Kapitanov, A., Kvanchiani, K., Nagaev, A., Kraynov, R., Makhliarchuk, A., 2022. HaGRID — HAnd Gesture Recognition Image Dataset. *arXiv*:2206.08219, 1-12.

LeCun, Y, Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, 86(11), 2278–2324.

Lahiani, H., Elleuch, M., Kherallah, M., 2015. Real time hand gesture recognition system for android devices. *15th International Conference on Intelligent Systems Design* and Applications (ISDA), 591–596.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., Dollár, P.,2017. Focal Loss for Dense Object Detection. *ICCV*, 2999-3007.

Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., Zhang, F., Chang, C., Guang, Y.M., Lee, J., Chang, W., Hua, W., Georg, M., Grundmann, M., 2019. MediaPipe: A Framework for Building Perception Pipelines. *arXiv*:1906.08172, 1-9.

Mallat, S., 1989. A Theory for Multiresolution Signal Decomposition: The Wavelet Representation. *IEEE Trans. Pattern Anal. Mach. Intell.* 11(7), 674–693.

Otsu, N., 1979. A Threshold Selection Method from Gray-Level Histograms. *IEEE Trans. Syst. Man Cybern*. 9(1), 62–66.

Papandreou, G., Zhu, T., Chen, L., Gidaris, S., Tompson, J., Murphy, K., 2018. PersonLab: Person Pose Estimation and Instance Segmentation. *ECCV*, 269–286.

Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You Only Look Once: Unified, Real-Time Object Detection. *CVPR*, 779–788.

Simonyan, K., Zisserman, A. 2014. Two-Stream Convolutional Networks for Action Recognition in Videos. *NeurIPS*, 27, 568–576.

Wu, Z., Sun, X., Jiang, H., Mao, W., Li, R., Andriyanov, N., Soloviev, V., Fu, L., 2023. NDMFCS: An automatic fruit counting system in modern apple orchard. *Comput. Electron. Agric.* 211:108036.

Zhang, H., Rao A., Agrawala, M., 2023. Adding Conditional Control to Text-to-Image Diffusion Models. *ICCV*, 3836-3847.

Zhang, H., Wang, C., Sun, Y., Dasgupta, E., Chen, H., Leonardis, A., Zhang, W., Jin H., 2023. Hybrid Pose and Context-based Gesture Recognition. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 17163-17173